The Two-Column Data Editor

Matthew Marcus
August, 2002

I. Introduction

This program allows editing and some mathematical operations on two-column (Y *vs*. X) data. The functions provided include plotting, math operations such as differentiation and resampling, background subtraction, pruning to a range, removal of points within a range, functions on one of the coordinates (i.e. $x \to x + \Delta x$ or $y \to \exp(y)$), and reading off of coordinates of points or extrema. Like `Plot-add-multiply`, and unlike most of the other code in this package, this program is not specialized to EXAFS or even X-ray data. Any Y *vs*. X data can be manipulated.

The operations are grouped into tabs, each containing controls for related operations. Some operations refer to one or more cursors. These are always enabled, though not always on-scale. There are buttons to bring them back. For those operations which refer to one cursor, the last one manipulated is the one used. For those involving two, it doesn't matter which one is which.

There is a multi-level Undo feature. Also, to help you keep track, the program displays a description of the last operation performed.

Instead of explaining all the functions one by one, I'll go through a worked example in which at least one function from each tab is used. The example is a raw EXAFS data file for which we want a pre-edge-subtracted, normalized curve suitable for XANES or extended XANES analysis. Thus, we want to:

1. Read in the input file.
2. Remove the pre-edge background
3. Find the edge energy and shift the scale so that it's at 0eV
4. Find a smooth curve which goes through the end of the data (post-edge region where the EXAFS is small) and divide by it.
5. Prune to a range of –50 to +100eV
6. Write the output file.

The net result is a curve which starts at 0, rises up at the edge, and oscillates around 1.

II. The example

We start by reading in a file. When the program starts up, it asks for a file. Also, you can use the Read New File button on the File I/O tab. The file is `crfoil.r`, and it's a transmission spectrum from Cr metal.

Next, use the graph tools to rescale the axes so we can see the whole graph. The result looks like Figure 1. Note that the Undo/Redo buttons are grayed out because nothing has yet been done. Again using the graph tools, we zoom in on the pre-edge region. We can use the type of zoom which just works on the X-axis, then hit the Rescale Y range button, and the Y scale will be redone to make the data within the given X range fit.

Now we switch to the Background tab. This tab has two controls, each of which can be set to two different modes. The Form of background control chooses what kind of background will be removed, whether it's linear or exponential. In this example, we'll use both. The Selection Mode control offers two choices. Defined by cursor creates a line or exponential curve which goes through the points defined by the two cursors. Fit data between cursors takes the set of data with abscissae lying between those of the two cursors and fits a line or exponential to it. For the pre-edge, we want to subtract a linear background, so we use linear for Form of background. We'll specify it manually. If the cursors aren't on screen, use the Bring buttons to bring each one on-screen. Now, the screen looks as in Figure 2. To subtract the background, hit the Subtract (–) button. The Divide by button (÷) would divide the data by the background. We'll want that later, but not now.

Next, we want to find the edge position and subtract that energy value from the abscissa, so that the edge appears at 0eV. One popular definition of edge position is the first major maximum in the derivative. We therefore go to the Math tab and select Differentiate. The resulting graph has many peaks, including a strong one at the edge. Zoom in on this (Figure 3.), then go to the Coordinates tab to see where it is. To find the position of an extremum, we bring a cursor to just before that position, then hit the Get Extremum button. The Coordinates tab has spaces for an X and a Y coordinate, called Delta X and Delta Y. The Get Extremum operation loads the coordinates of the found extremum into the Delta X and Delta Y controls and puts the cursor at that point.

We are now at the point shown in Figure 4. The Trade X/Y button is missing from this picture; it was just added during the writing of this manual.

Now, if we wanted to shift the curve that's on-screen by the found edge energy (Delta X), we would just hit X->X-DeltaX on the Coordinates tab. This would subtract the edge energy from the abscissa for the curve being displayed. However that curve isn't the one we want. Fortunately, the one we want is an Undo away, since the last operation that did anything to the curve was the Differentiate. Thus, we hit Undo, then X->X-DeltaX, and that shifts the pre-edge subtracted data leftwards by the edge energy. The result, after using the rescale buttons on the graph, is shown in Figure 5.

To get a normalized curve, we want to divide the present curve by something plausible and smooth which represents the smooth part of the absorption. This is of course what the EXAFS analysis routine does, but here we want a smooth curve without many degrees of freedom, so that we preserve the shape of the edge. Since the post-edge is mildly curved, we can use an exponential to simulate it. Thus, we go back into the Background tab and this time choose Exponential and Fit between cursors. This step is shown in Figure 6. The red line is the background. Note that we have chosen the data range so that the red curve goes through the EXAFS wiggles at the start. This was an attempt to keep the EXAFS wiggles from biasing the fit. The right-hand cursor is clipping a little just so you can see it. Now we hit the Divide by button to normalize. The result now oscillates around 1 above the edge.

Let's suppose that we only want the data from 50eV below the edge to 100eV above. We go to the Cuts tab, set the cursors to the desired range, and hit Delete outside cursors. The data are pruned to just the set of points lying within the cursor abscissa range. The ordinates on the cursors don't matter. The result is shown in Figure 7.

Finally, we can write the results using the Write File button on the File I/O tab. The results being written are as shown in Figure 8.

III. Function reference

This section lists all of the functions, grouped by tab.

(no tab)	Undo	Undoes the last operation. It's multilevel.
	Redo	Undoes the Undo. These buttons are grayed and disabled when there's nothing to Un/Redo.
	Bring	Brings a cursor on-screen.
	Rescale Y range	Sets the Y range so that all data within the displayed range of abscissa are displayed.
	Stop	Stops the program
File I/O	Read New File	The obvious. There is no default extension.
	Write file	Again, obvious. Two-column ASCII format is used for all file I/O.
Coordinates	DeltaX	An X-coordinate or operand for transformations on the abscissa. You can type in a value, load it from a cursor, or use Get Extremum.
	DeltaY	Same, for the ordinate.
	Load X of cursor	Loads into DeltaX the abscissa value for the active cursor, i.e. the one which was last moved or manipulated.
	Load Y of cursor	Same for DeltaY and ordinate.
	Get Extremum	Finds the first extremum of Y for X > the present active cursor abscissa. Fits a parabola through three adjacent points to find an X and Y value. Loads these into DeltaX and DeltaY.
	Trade X/Y	Interchanges X and Y in data.
	X<-ln(X),Y<-ln(Y) X<-X+DeltaX X<-X-DeltaX, etc.	Performs the specified math operation on the appropriate coordinate of the data.
Background	Subtract(−)	Subtracts specified background (see below) from data.
	Divide by(÷)	Divides data by specified background. These two green buttons are the only ones on this tab which do anything to the data; the others help specify what's to be done.
	Form of background	Specifies that the background is a linear or exponential function. Set this, the Selection mode, and the cursor positions before pushing the Subtract or Divide by buttons.

	Selection mode	If this is Fit data between cursors, the background curve is a least-squares fit to the data with abscissae between those of the cursors. If it's Defined by cursors, the background curve goes through the two points defined by the cursors.
Math	Differentiate	Runs a spline through the data and takes the the derivative. The data must be sorted in X. The derivatives at the end are estimated by running a cubic through the first and last four data points.
	Integrate	Same, but integrates the spline instead of differentiating. The integral is 0 for the first point.
	Sort Ascending	Sorts the data in order of increasing X. You have to do this in order to get the other buttons to work if the data aren't sorted.
	Resample	Spline interpolates onto a uniform grid containing a user-specified number of points.
	Remove redundant points	If X repeats, say because the monochromator stuttered, this function consolidates the points into one whose coordinates are the average of those of the group of points with the same X. Tolerance is how far apart the X coordinates have to be to be considered different.
Cuts	Delete betw. cursors	Deletes all points with X between the X values of the cursors.
	Delete outside cursors	The compliment – keeps only points with X between the X values of the cursors.

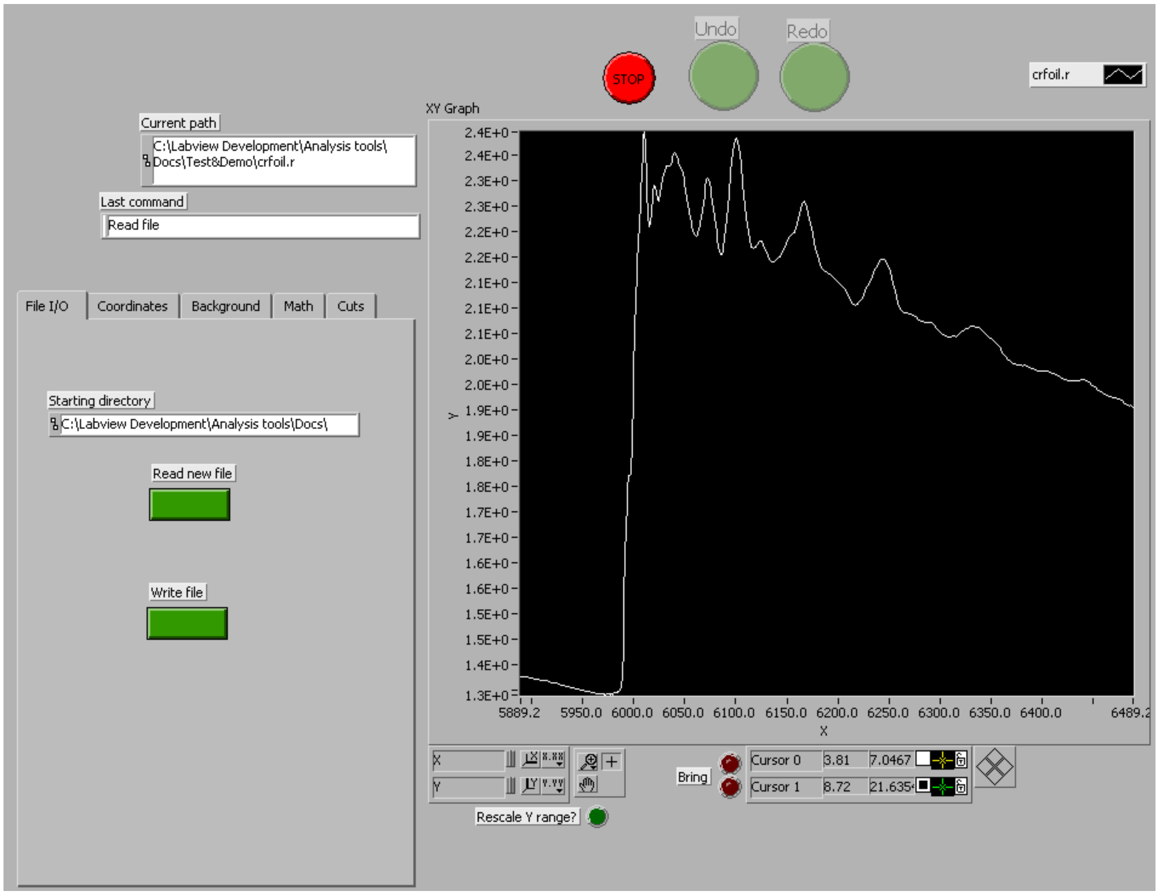

Figure 1. The program after reading in the input file and rescaling with the graph buttons. No cursors show because they're off-scale, but can be recalled with the Bring buttons. The I/O *tab is showing. Note that the file path is shown in the* Current Path *indicator.*

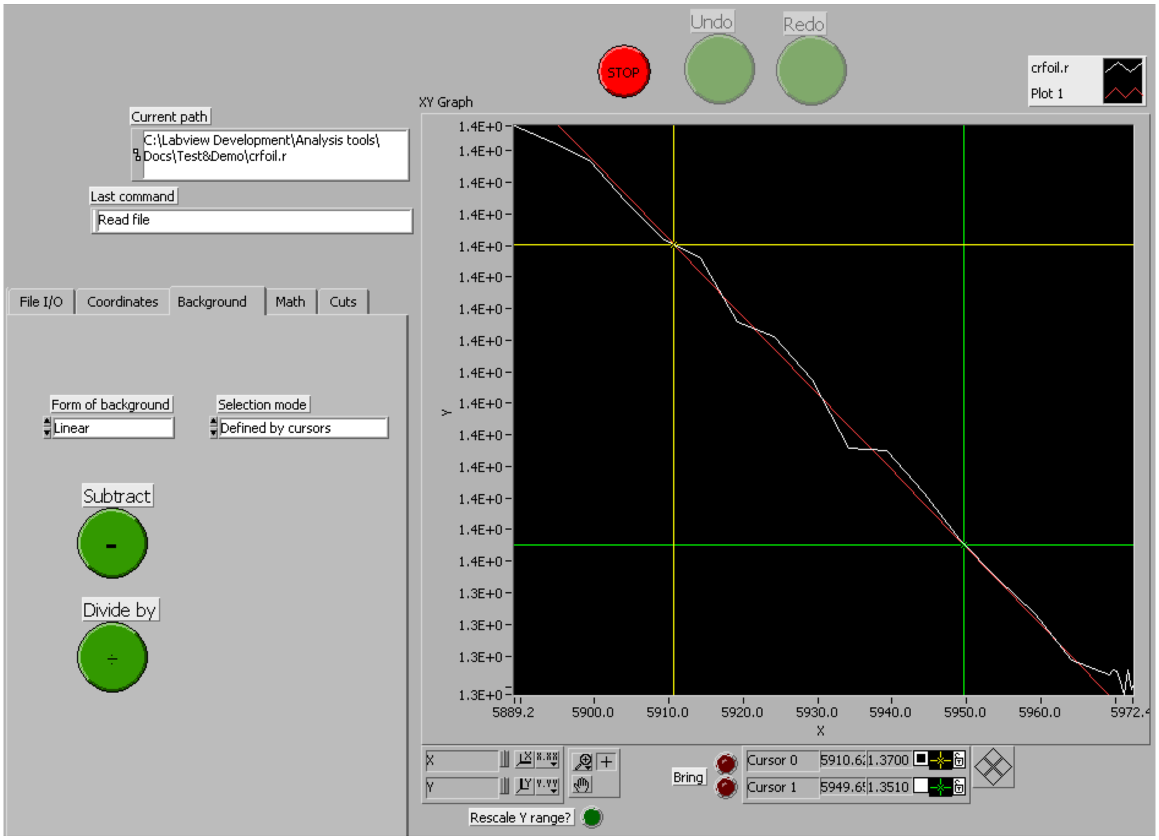

Figure 2. We've zoomed into the pre-edge region and defined a linear background which will be subtracted off when the Subtract *button is pushed.*

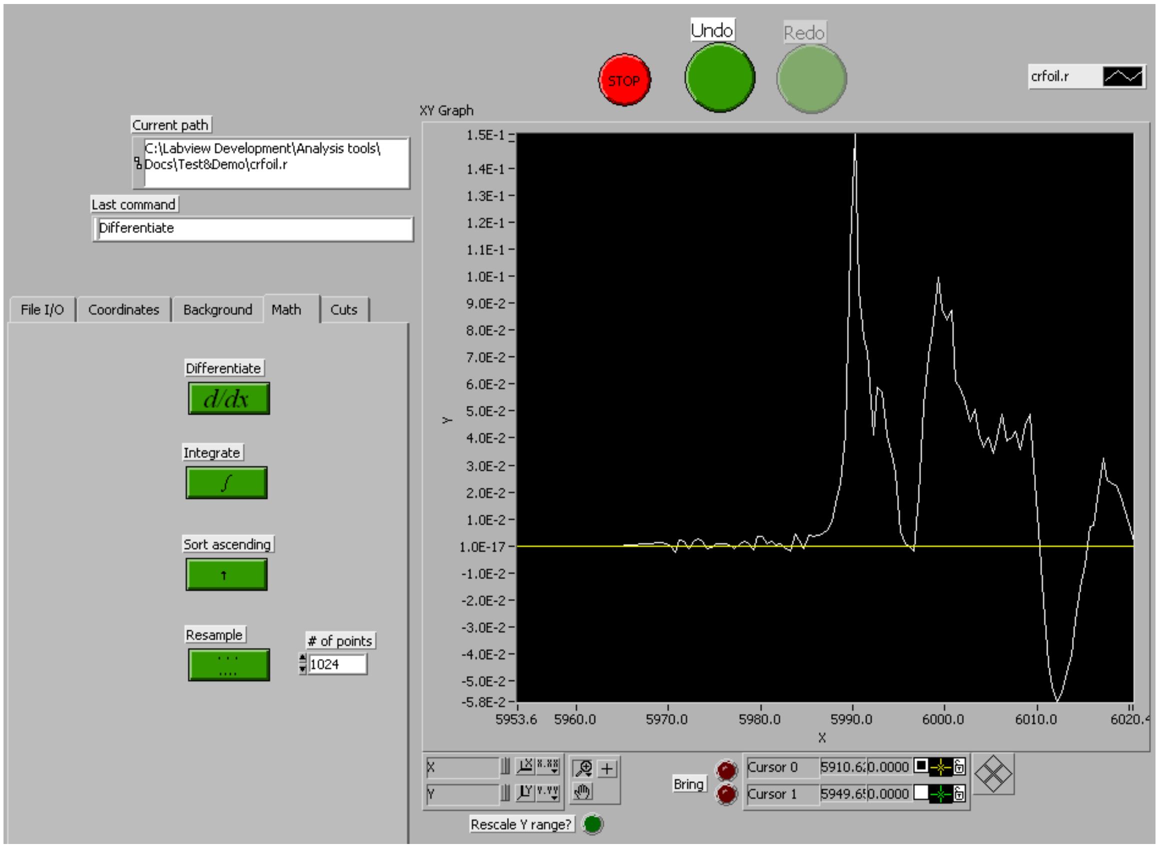

Figure 3. The Math tab includes the Differentiate *function which was used to differentiate the graph, resulting in the curve shown here after zooming in.*

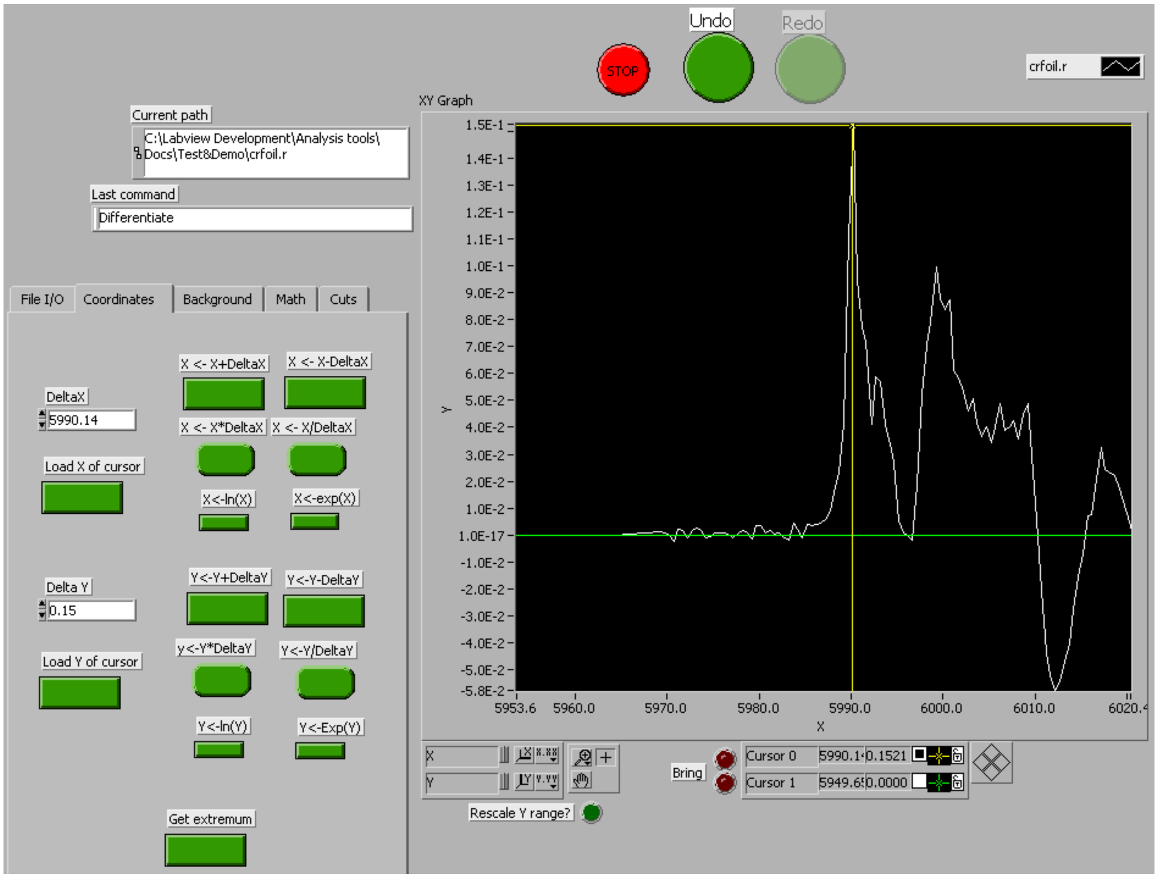

Figure 4. Using the Get Extremum *button on the* Coordinates *tab, we've found the peak position. The coordinates of that peak are loaded into the active cursor and into the* DeltaX *and* DeltaY *controls.*

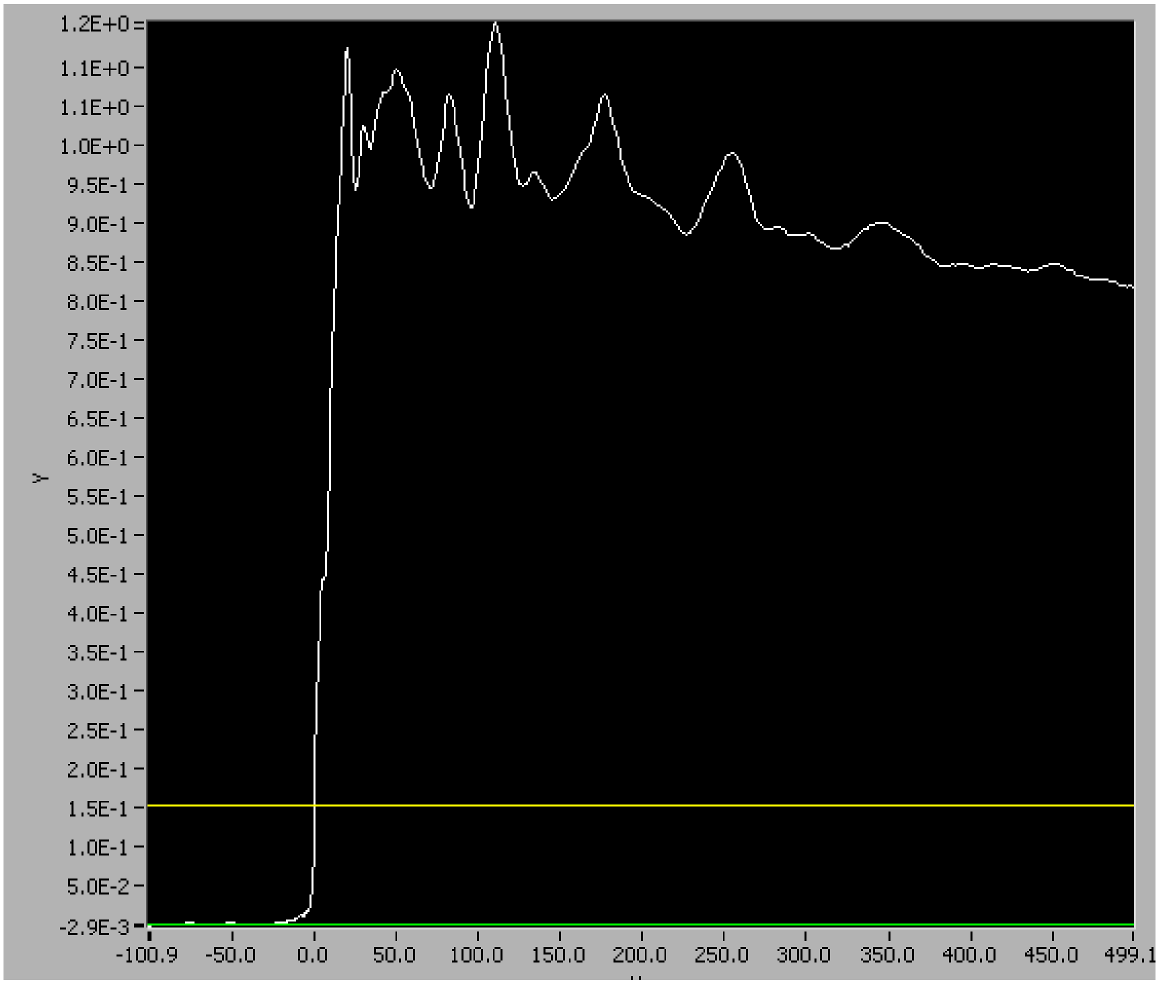

Figure 5. The curve after pre-edge subtraction and shifting X by the DeltaX *value found using the* Math *tab (Figure 4).*

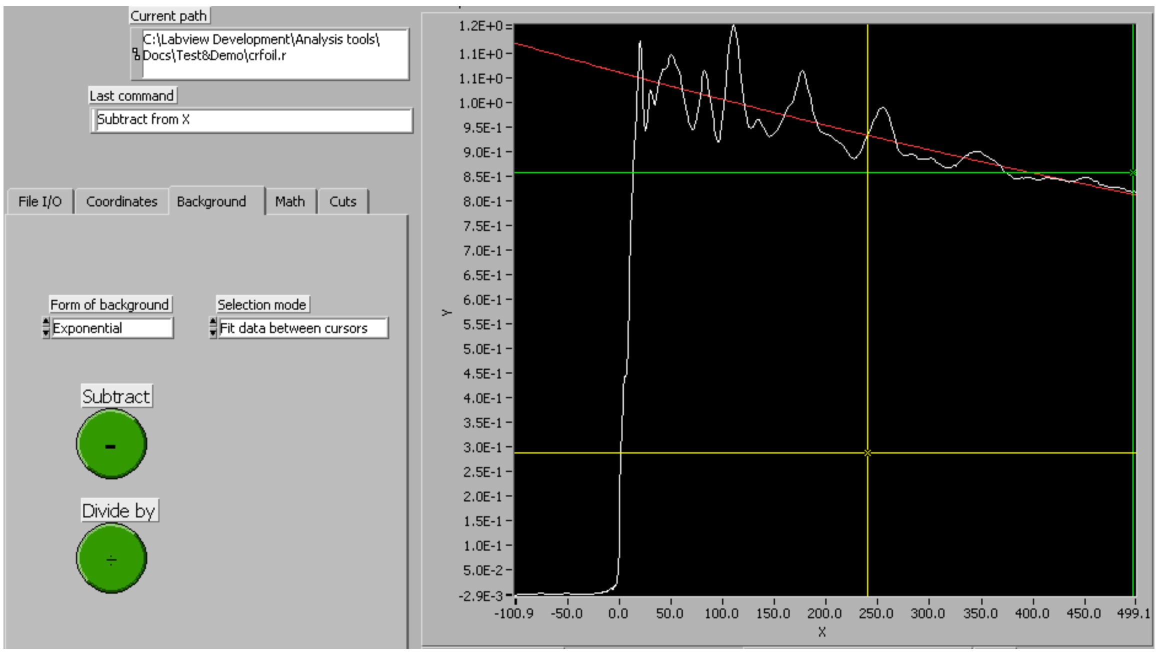

Figure 6. Running an exponential background through the data, fitted to the last half of the post-edge. Pushing the Divide by *button will normalize the data.*

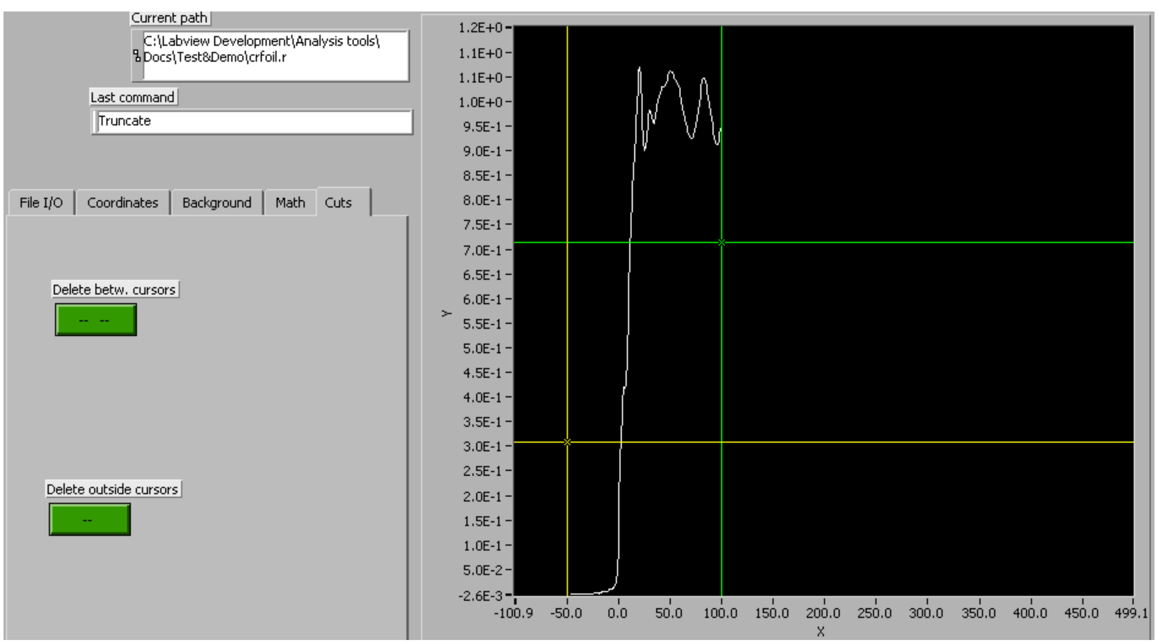

Figure 7. Picking out a range using the Delete outside cursors *button on the* Cuts *tab.*

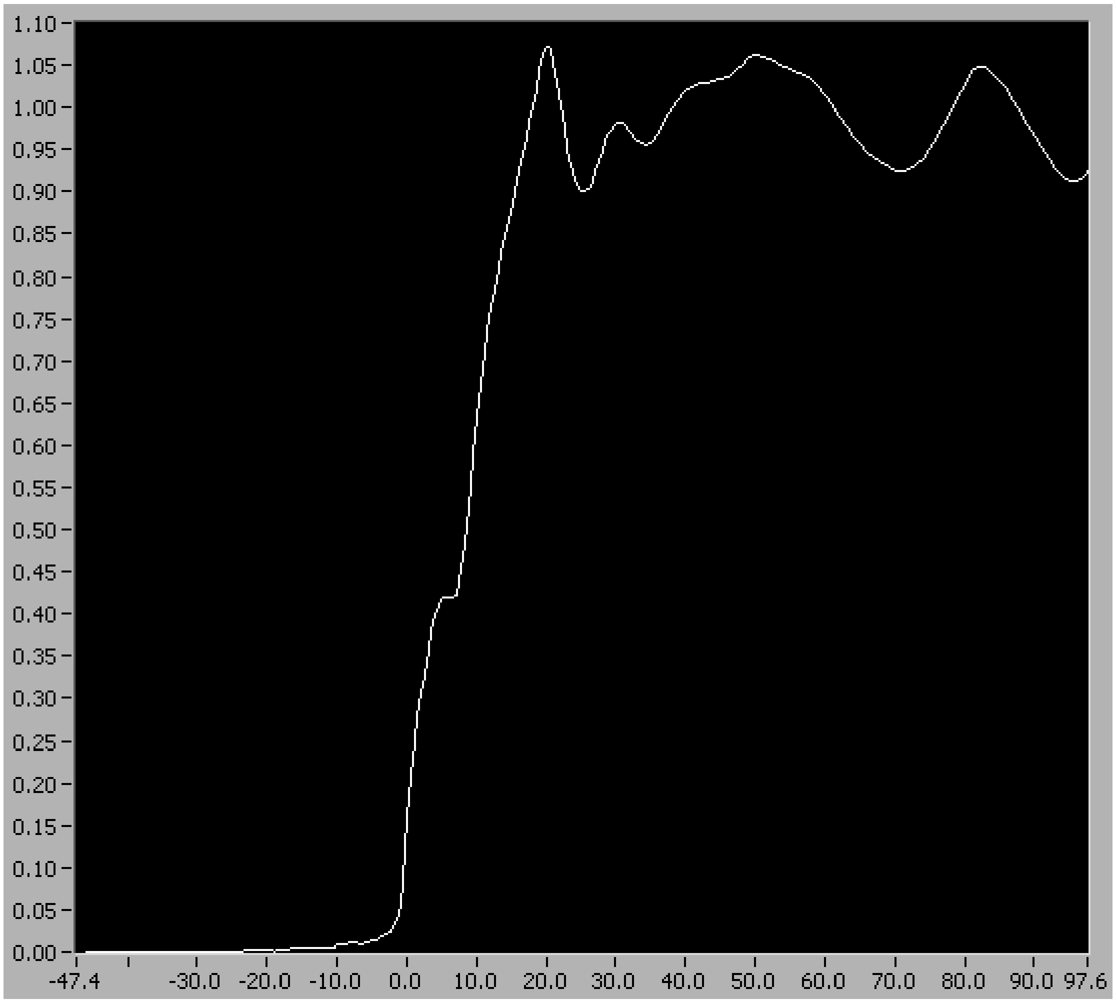

Figure 8. The final output, which can be written to a file and used elsewhere.